

Amendment to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Claim 1 (currently amended): A method of analyzing multi-threaded programs, comprising:

determining that unsynchronized accesses to a resource of interest can be performed by a plurality of threads;

receiving a request from a first thread to access the resource, ~~wherein the resource is available;~~

suspending the first thread for requesting to access accessing the resource to which unsynchronized accesses can be performed ~~that is available~~; and

while the first thread is suspended, receiving a request from a second thread to access the resource.

Claim 2 (original): The method of claim 1, wherein the requests of the first and second threads is to write data to the resource.

Claim 3 (original): The method of claim 1, further comprising awakening the first thread.

Claim 4 (original): The method of claim 1, further comprising logging for a user that the first and second thread performed unsynchronized accesses to the resource.

Claim 5 (original): The method of claim 1, wherein the first thread is suspended for a predetermined time, meaning that the first thread awakens after the predetermined time expires.

Claim 6 (previously presented): The method of claim 5, wherein the first thread is also suspended on an event, meaning that the event awakens the first thread.

Claim 7 (original): The method of claim 6, wherein the second thread sends the event that awakens the first thread.

Claim 8 (original): The method of claim 1, wherein the resource is a memory location, region of memory, hardware component, or peripheral device.

C) Claim 9 (currently amended): A computer program product for analyzing multi-threaded programs, comprising:

computer code that determines that unsynchronized accesses to a resource of interest can be performed by a plurality of threads;

computer code that receives a request from a first thread to access the resource, ~~wherein the resource is available~~;

computer code that suspends the first thread for requesting to access ~~accessing~~ the resource to which unsynchronized accesses can be performed ~~that is available~~;

computer code that while the first thread is suspended, receives a request from a second thread to access the resource; and

a computer readable medium that stores the computer codes.

Claim 10 (original): The computer program product of claim 9, wherein the computer readable medium is selected from the group consisting of CD-ROM, floppy disk, tape, flash memory, system memory, hard drive, and data signal embodied in a carrier wave.

Claim 11 (currently amended): A method of analyzing multi-threaded programs, comprising:

determining that unsynchronized accesses to a memory location can be performed by a plurality of threads;

receiving a request from a first thread to write data to the memory location, ~~wherein the memory location is available for writing~~;

suspending the first thread for requesting to write ~~writing~~ data to the memory location to which unsynchronized accesses can be performed ~~that is available for writing~~; and

while the first thread is suspended, receiving a request from a second thread to write data to the memory location.

Claim 12 (original): The method of claim 11, further comprising awakening the first thread.

Claim 13 (original): The method of claim 11, further comprising logging for a user that the first and second thread performed unsynchronized writes to the memory location.

Claim 14 (original): The method of claim 11, wherein the first thread is suspended for a predetermined time, meaning that the first thread awakens after the predetermined time expires.

Claim 15 (previously presented): The method of claim 14, wherein the first thread is also suspended on an event, meaning that the event awakens the first thread.

Claim 16 (original): The method of claim 15, wherein the second thread sends the event that awakens the first thread.

Claim 17 (currently amended): A computer program product for analyzing multi-threaded programs, comprising:

computer code that determines that unsynchronized accesses to a memory location can be performed by a plurality of threads;

computer code that receives a request from a first thread to write data to the memory location, ~~wherein the memory location is available for writing~~;

computer code that suspends the first thread for requesting to write writing data to the memory location to which unsynchronized accesses can be performed that is available for writing;

computer code that while the first thread is suspended, receives a request from a second thread to write data to the memory location; and

a computer readable medium that stores the computer codes.

Claim 18 (original): The computer program product of claim 17, wherein the computer readable medium is selected from the group consisting of CD-ROM, floppy disk, tape, flash memory, system memory, hard drive, and data signal embodied in a carrier wave.

Claim 19 (currently amended): A method of analyzing multi-threaded programs, comprising:

determining that unsynchronized accesses to a memory location can be performed by a plurality of threads;

receiving a request from a first thread to write data to the memory location, wherein the memory location is available for writing;

suspending the first thread for requesting to write writing data to the memory location to which unsynchronized accesses can be performed that is available for writing;

while the first thread is suspended, receiving a request from a second thread to write data to the memory location;

awakening the first thread; and

logging for a user that the first and second thread performed unsynchronized writes to the memory location.

Claim 20 (original): The method of claim 19, wherein the first thread is suspended for a predetermined time, meaning that the first thread awakens after the predetermined time expires.

Claim 21 (previously presented): The method of claim 20, wherein the first thread is also suspended on an event, meaning that the event awakens the first thread.

Claim 22 (original): The method of claim 21, wherein the second thread sends the event that awakens the first thread.

Claim 23 (currently amended): A computer program product for analyzing multi-threaded programs, comprising:

computer code that determines that unsynchronized accesses to a memory location can be performed by a plurality of threads;

computer code that receives a request from a first thread to write data to the memory location, wherein the memory location is available for writing;

computer code that suspends the first thread for requesting to write writing data to the memory location to which unsynchronized accesses can be performed that is available for writing;

computer code that while the first thread is suspended, receives a request from a second thread to write data to the memory location;

computer code that awakens the first thread;

computer code that logs for a user that the first and second thread performed unsynchronized writes to the memory location; and

a computer readable medium that stores the computer codes.

Claim 24 (original): The computer program product of claim 23, wherein the computer readable medium is selected from the group consisting of CD-ROM, floppy disk, tape, flash memory, system memory, hard drive, and data signal embodied in a carrier wave.

Claim 25 (currently amended): A method of analyzing multi-threaded programs, comprising:

modifying an existing multi-threaded program include computer code that determines that unsynchronized accesses to a memory location can be performed by a plurality of threads;

modifying the existing multi-threaded program to include computer code that suspends a first thread for requesting that requests to write data to a memory location to which unsynchronized accesses can be performed that is available, wherein a second thread writes data to the memory location; and

modifying the existing multi-threaded program to include computer code that logs for a user that the first and second thread performed unsynchronized writes to the memory location when a second thread writes data to the memory location.

Claim 26 (original): The method of claim 25, wherein the first thread is suspended for a predetermined time, meaning that the first thread awakens after the predetermined time expires.

Claim 27 (previously presented): The method of claim 26, wherein the first thread is also suspended on an event, meaning that the event awakens the first thread.

C | Claim 28 (original): The method of claim 27, wherein the second thread sends the event that awakens the first thread.

Claim 29 (currently amended): A computer program product for analyzing multi-threaded programs, comprising:

computer code that modifies an existing multi-threaded program include computer code that determines that unsynchronized accesses to a memory location can be performed by a plurality of threads;

computer code that modifies the existing multi-threaded program to include computer code that suspends a first thread for requesting that requests to write data to a memory location to which unsynchronized accesses can be performed that is available, wherein a second thread writes data to the memory location;

computer code that modifies the existing multi-threaded program to include computer code that logs for a user that the first and second thread performed unsynchronized writes to the memory location when a second thread writes data to the memory location; and

a computer readable medium that stores the computer codes.

Claim 30 (original): The computer program product of claim 29, wherein the computer readable medium is selected from the group consisting of CD-ROM, floppy disk, tape, flash memory, system memory, hard drive, and data signal embodied in a carrier wave.